奔流 Flows

自由、奔放的技术刊物

本期推荐：无处不在的 Troubleshooting

网络版：http://www.netis.com.cn/flows

2012 年 08 月 01 日 第 34 期

《奔流》杂志 Flows Magazine | Netis Discover IT, Value IT, Love IT.

奔流 Flows

自由、奔放的技术刊物

主办　上海天旦网络科技发展有限公司
Netis Technologies Co., LTD

总编辑　Nancy Yang
Nancy.yang@netis.com.cn

特约供稿人　Neil Zhang
neil.zhang@netis.com.cn
Charlie Liu
charlie.liu@netis.com.cn
Simon Fu
simon.fu@netis.com.cn

联系地址　上海市浦东新区浦建路 76 号
由由国际广场 23 层
电话　021-58702300
传真　021-58702766
投稿地址　flows@netis.com.cn
出版日期　2012 年 8 月 1 日
公司网址　http://www.netis.com.cn
奔流网址　http://www.netis.com.cn/flows
技术采编源　网络分析专家论坛
http://www.netexpert.cn

目录 2012 年 8 月刊

卷首语　永不停歇的变革

新闻速递　NetScout System 公司公布 2013 财年第一季度财务状况
VSS Monitoring 引入全新网络数据包中转设备
实现全方位可视性，提高运行性能
全球超过 275 所大学使用 Splunk 助其智能运营

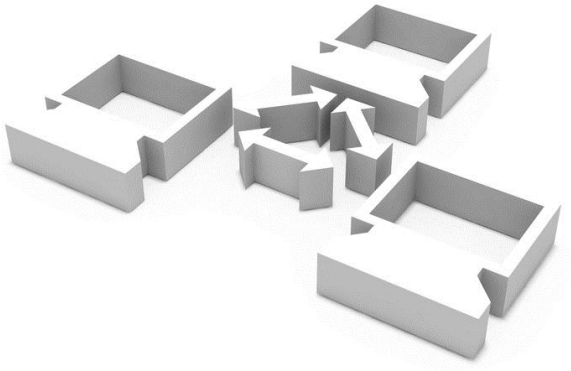
技术关注　无处不在的 Troubleshooting
网络虚拟化——互联网的下一波革命
轻松分析 DHCP

TCP/IP　TCP 滑动窗口简介
TCP/IP 知识你问我答

Netis 正在招聘　Netis 目前有若干开放的职位，欢迎大家应聘或推荐朋友。详细信息可以查阅 Netis 在 智联招聘上发布的职位 或请联系 recruit@netis.com.cn

投稿说明　尽管只是一个内部期刊，我们欢迎任何人的投稿，对于任何被选中发表的内容，Netis 都会按国家标准支付稿费表示感谢。《奔流》刊登内容的主要标准，是基于技术的准确性、内容的关联程度或具备独到的观点与视角。任何类型希望增强改善这本小杂志的想法也非常欢迎。联系投稿请至 flows@netis.com.cn 或登录 http://www.netis.com.cn/flows 了解更多。

永不停歇的变革



Nancy

我们所生活的世界，从来都不会和想象中那样简单。从人类开始直立行走的那一刻开始，就开始了永不停歇的变革活动。

在历史的漫漫长河里，或剧烈或微弱的变革都在时时刻刻推动世界沿着阶梯向更进步，更伟大的目标进取和攀爬。而世界范围内不断来袭的政治体制、经济体制、社会体制、生产力、宗教或是思想文化等一系列的变革，刷新着旧的观念和体制，带来了全球新规则的诞生。

民主、经济和技术等活跃的社会力量，促进着人类的不断进步。这些力量的深远意义完全可以和历史上任何一个巨大的变革时期相提并论。

如今的变革，变得越来越迅速。我们正在尝试了解生命的起源，掌握人类的基因治疗疾病；我们正在尝试用最伟大的技术和思想，创造运转速度更快，存储量更大，运算水平更高的信息系统；我们正在尝试实现全球的信息资源共享，我们正在努力于全球人类无地域约束的轻松交流方式。让人类在现代化科技的浪潮中更加高效，快捷，愉悦，成为了这个时代的美好愿景与奋斗方向。

不断变革，不断探索和理解，这是人类的职责。

NetScout System 公司公布 2013 财年第一季度财务状况

2012 年 7 月 19 日致力于提供先进应用和服务保证解决方案的行业领导者 NetScout Systems Inc.（NASDAQ：NTCT）公布了截止于 2013 年 6 月 30 日的 2013 财年第一季度财报。

	Q1 FY 2013	
	GAAP	**Non-GAAP**
Revenue (in millions)	$76.4	$76.5
Net income (in millions)	$5.0	$8.1
Net income per share	$0.12	$0.19

“在第一季度，收益和 EPS 数据都表现出众，”NetScout Systems 公司董事长兼 CEO Anil Singhal 先生表示。“产品总预订量同比增长 27%，这样的业绩振奋人心。同时，我们很高兴地在今天宣布，NetScout 已收购了 Accanto Systems 公司。Accanto Systems 公司致力于电信服务供应商的服务保障。Accanto Systems 带来的新技术与 NetScout 的 packet flow 策略结合，可为下一代网络语音服务，包括 VoIP 和 VoLTE 带来更好的语音服务监测性能。”

VSS Monitoring 引入全新网络数据包中转设备

实现全方位可视性，提高运行性能

2012 年 7 月 24 日，全球网络数据包中转（Network Packet Brokers ，NPB）领域的领导者 VSS Monitoring 公司今日推出全新 NPB 设备与软件套件，可为网络监控和安全系统提供全方位可视性，提高运行性能。

当今，规模日益庞大的监测和安全系统让网络变得越来越复杂，这也给企业和服务供应商带来了诸多运营与服务的挑战，网络可视性能有限、投资费用固化、复杂的网络设备或软件的升级都限制了现有工具或系统的运行速度和性能。

“网络复杂性与升级成本日趋加剧，市场迫切需要一种全新的应用感知网络数据包中转设备，”Gartner 研究主管 Jonah Kowall 表示。“网络数据包中转设备使得管理当今快速增长的数据流成为可能。在网络数据包中转设备的帮助下，可以更快捷地部署并运行监测设备，实现更佳的可视性能。”

全球超过 275 所大学使用 Splunk 助其智能运营

2012 年 7 月 24 日报道，业界领先的实时智能运营软件供应商 Splunk 公司长期关注着全球大学园区的运营改善，特别是 IT 基础架构、应用管理以及安全领域。全球超过 275 所大学都在使用 Splunk 帮助其实时分析海量的设备数据，提供关键视图，快速识别潜在威胁。近期，正在使用 Splunk 的大学有：澳大利亚墨尔本大学、伯明翰 Alabama 大学、Lone Star 大学以及佛吉尼亚理工大学等。

无处不在的 Troubleshooting

By Neil

18 世纪 60 年代第一次工业革命以蒸汽机的改良和广泛应用使人类用机器生产代替手工劳动进入了蒸汽时代；19 世纪 70 年代第二次工业革命以电力的发现和广泛应用使人类从蒸汽时代进入电气时代；20 世纪四五十年代第三次工业革命以原子能、计算机、航天技术、生物工程等领域取得重大突破使人类进入了信息时代；2012 年 Netis 自主研发的 Crossflow NPM 助力 IT 的网络部门，从网络层面故障诊断的手工劳动逐步进入了自动化运维时代，Netis 从来没有想过要与工业革命媲美，但 Netis 的自主创新精神与工业革命却不谋而合。

Netis 一直致力于将 TCP/IP 的 Troubleshooting 变得更加简单,我们在实践中不断的成长，长期致力于在各大运营商、银行、证券行业，不断做着各种具有挑战性的 Troubleshooting，当然，不仅仅是网络故障，还包括应用或者主机故障。

当年我在考 CCIE 的西天取经途中，经常听到老师或者师兄弟讲，如果你会配置简单的路由交换协议，搞定低端的交换机，路由器，那么你算是一个合格的 CCNA。如果你会玩 BGP、MPLS、VPN，搞定 65、76 级别设备，那么你算是一个合格的 CCNP。如果前面的活你同时能干好，而且你还能在设备上把 Debug 玩的很熟练，能够处理各种复杂的网络出现的异常，不管是 BGP 也好，组播也罢，那么你算是一个合格的 CCIE 了。问题来了，如果某个客户端到某个应用主机的访问有少量的丢包，你怎么办？如何确定是网络层设备丢包还是主机侧没有给予响应。有人说我在我的网络设备一路 ping 应用主机有没有丢包就好了，路由器当前理论值支持 ping 的精度达到 1ms 发出一个 ping 包，即 1S 发出 1000 个数据包，我曾经测试过如果你把发出 ping 包的精度设置到 1ms，实际测试的结果要远远小于 1000 这个数值。而且对于 1000M 端口的网络设备而言，1S 处理上百万个数据包只是家常便饭，更别说当前华为发布 CloudEngine 12812 交换机可支持的 96 个 100G 以太网端口，1S 发出的数据包数量那就是天文数字。但是当把这些问题交给 Netis 的 Crossflow NPM 就好办了。

Netis 的工程师不光只是在客户那里做着各种各样复杂万千的 Troubleshooting，在 Netis 的 Office 里，我们也会分析当前自己遇到的各种问题，下面列举一二以做抛砖引玉：

一、 登录 web 页面被弹出

登录某 web 页面进入后，约 5S 后被弹出界面，于是开启 Sniffer Portable 进行抓包分析，定位精准的时间点：

- 时间：2012-05-09
- 故障现象：登录 web 界面 5s 左右弹出
- 故障次数：2 次
  - 9:45 登录被弹出
  - 9:50 登录被弹出

分析步骤：用 Sniffer Portable 进行源和目的 IP 地址的过滤分析 172.16.26.153→172.16.11.161

9:45 过滤的数据包，如图 1-1。9:50 过滤的数据包，如图 1-2。

从上面过滤出来的两个数据包，我们很容易就找到了登录后被 web 页面弹出问题的原因，从而得出分析结论：

1、数据包显示两次流量网页未被授权 Unauthorized 访问；

2、显示 HTTP 401 错误，导致 web 访问被弹出，需要重新登录。

我们再 Google 搜索一下就知道有哪些原因会导致未被授权访问了，并且还给出了对应的解决方法。

a)、错误号 401.1 未经授权：访问由于凭据无效被拒绝。

b)、错误号 401.2 未经授权：访问由于服务器配置被拒绝。

c)、错误号 401.3 未经授权：访问由于 ACL 对所请求资源的设置被拒绝。

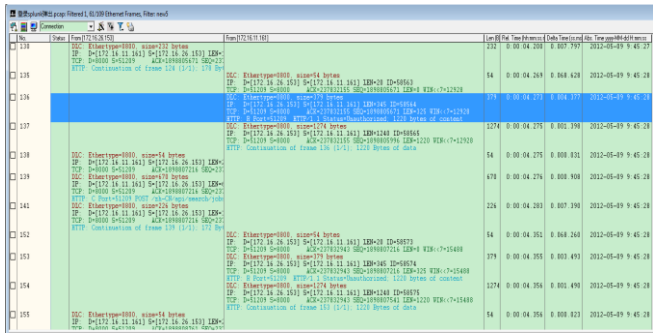


图 1-1

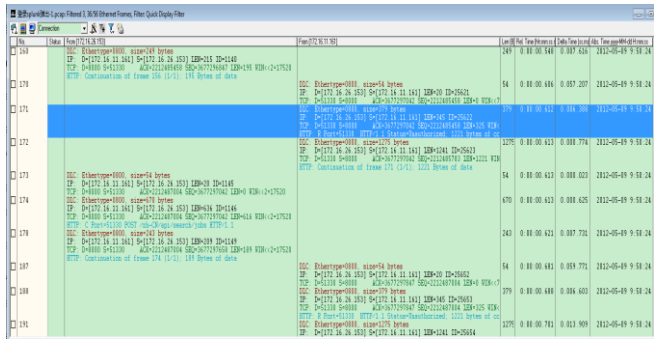


图 1-2

图 1-3

图 1-4

二、Solarwinds 网管软件报交换机重启

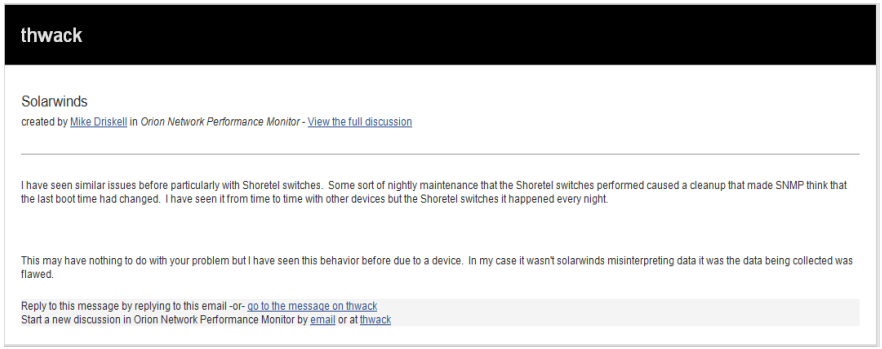
某日 Solarwinds 网管软件发出 Alert 交换机发出重启的告警信息，但是 telnet 登录到交换机，发现交换机并没有重启，没有任何重启的 log 信息，而且显示该交换机的 uptime 时间已经是运行了 118weeks。

- 时间：2012-5-18
- 12:47:35：Solarwinds 报交换机重启的 Alert
- 登录交换机检查设备没有重启

分析步骤：从 Netscout 的 Infinistream 提出交换机与 Solarwinds 访问的数据包，过滤出源和目的 ip 地址。

过滤数据包，如图 1-3。从 12:47 和 12:49 的数据包对比看出交换机 Uptime 时间从 2863305004ticks 减少到了 5600ticks，意味着交换机只是运行了 56S（1S=100ticks）。如图 1-4。

从数据包层面我们看出交换机在 12:47 发出的系统启动时间 Uptime 是 2863305004ticks 的时间，我们再来看一下什么是 CPU Ticks 时间： CPU 的 ticks 时间计算方法 100ticks=1s，Ticks 时间的范围 0~2*32-1（4294967295），过了这个值之后会重新计数（大概是 497 天）而此时交换机运行时间 2863305004 ticks< 4294967295 ticks。设备正常工作时 CPU ticks 的时间随运行时间的增加而不断增加，如果设备重启，那么 CPU ticks 会重新从 0 开始计算。


thwack

Solarwinds

created by Mike Driskell in *Orion Network Performance Monitor* - View the full discussion

I have seen similar issues before particularly with Shoretel switches. Some sort of nightly maintenance that the Shoretel switches performed caused a cleanup that made SNMP think that the last boot time had changed. I have seen it from time to time with other devices but the Shoretel switches it happened every night.

This may have nothing to do with your problem but I have seen this behavior before due to a device. In my case it wasn't solarwinds misinterpreting data it was the data being collected was flawed.

Reply to this message by replying to this email -or- go to the message on thwack
Start a new discussion in Orion Network Performance Monitor by email or at thwack

分析结论：交换机的 CPU ticks 时间出现了减少（在设备没有重启的情况下）异常，交换机主动发出了标示系统运行时间减少的 Uptime 时间的报文，此时 Solarwinds 收到该数据包即认为该设备出现重启的现象。

后续：我将该事件发在了 Solarwinds 的官方论坛里以便同大家一起探讨一下，得到了 Solarwinds 官方的积极响应。诊断结论也不谋而合。

在生活中，我们也经常会面临着各种各样的棘手问题需要去处理，但是我想只要我们把它分析的像数据包一样,理解每一个字段的内容含义,做到多段事情的关联分析，也许事情会简单很多。

网络虚拟化——互联网的下一波革命

By 36 氪

最近几天开放网络峰会（Open Networking Summit）的召开掀起了一股软件定义网络（SDN）和 OpenFlow 的报道热潮。很多人相信，软件定义网络和 OpenFlow 主导的网络虚拟化必将引发互联网的下一场革命。本文介绍这一行业的领军者之一，Nicira。其 CTO Martin Casado 是 OpenFlow 协议的第一份草案的撰稿人。

成立于 5 年前的 Nicira 被认为是硅谷最吸引人的初创企业。简单来说 Nicira 是一家网络虚拟化公司，它的任务是把内核从网络硬件移植到软件中。这家公司的使命是建设只存在于软件之中的网络，使得底层的网络硬件对于网络用户来说不可见。从而让网络的组建、改造、重建都易如反掌，最最终支撑 web 上最大规模的服务（想想物联网的规模）。

其 CTO 为 Martin Casado，此人聪颖过人，是 OpenFlow 第一份规范草案的拟定者。Martin Casado 的愿景是网络可以像计算机一样可编程。

他认为既然谁都可以买一堆计算机然后投入一群软件工程师去做出一些了不起的东西，那么网络也理应如此。理想的网络架构应该具备计算机的灵活性，可适用于任何网络硬件，换句话说，应该有一种通用网络操作系统，在此平台下，硬件只负责收发网络数据包，由软件负责思考。

Nicira 的平台用户包括 AT&T、eBay、NTT、金融巨头 Fidelity 以及仅次于 Amazon 的云服务供应商 Rackspace。还有一些虽不具名但绝对是大名鼎鼎的互联网公司。

为什么这些公司愿意跟 Nicira 合作？因为当今的硬件网络的改造难度已经到了荒谬的地步。Yahoo 的前 CTO Raymie Stata 把复杂的计算机网络比作是 15 Puzzle 游戏（有点类似华容道，必须在 16 个格子的正方形中重新安排 15 个方格块），一旦要对网络做出调整，就得就硬件进行物理性地重新安排。

Nicira 的做法就是网络虚拟化，用软件实现网络调整，底层的硬件引擎无需改动。其实就是把交换机和路由器里面的智能部分取出移到软件上，这样交换机就不需要知道太多东西。Nicira 自 2009 年以来就一直跟 Rackspace 在合作，为后者的一项云服务新的 beta 版提供帮助。使得网络架构师把云服务的大权让位给了云架构师。

传统网络的问题：有史以来最安全网络的麻烦

Martin Casado 曾跟一家不具名的美国情报机构工作过，该机构的网络被他认为是有史以来最安全的计算机网络。但该网络的麻烦是建设这样的网络近乎不可能，而且一旦想要改动网络，同样的问题还会再次出现。

这个事件给 Casado 的启示是市场的力量缺位，尽管政府有钱却买不到想要的产品—安全和易于管理的网络。他举了个例子，改动一台计算机就得进行 8 项不同的配置变更，所以基本上这样的网络搭建好之后什么都不能动。

网络设备制造商不允许对硬件进行重新编程，代码都是直接写进交换机或着路

由器的。当然，他们这么做也是由原因的，用户买一个东西回来当然希望插上电就能用，另外网络公司也不希望因为用户修改而导致网络瘫痪。但是这种限制会给网络的扩容改造带来大问题。而对于业务规模不断扩大的大型互联网公司来说，这样的情况是常有的事。

Google 从 2005 年开始甚至开始自己造网络硬件，部分原因就是由于它需要对硬件能够有更多的控制。服务器之间需要高带宽的连接，同时网络还需要能够伸缩自如。传统的网络设备供应商做不到这一点，成本过高，系统太封闭，以 Google 的这种网络规模是其无法管理的。

所以 Martin Casado 到了斯坦福大学研究生院后就开始着手建设一种灵活的、能够像计算机一样可编程的网络。

通心面式代码之死

Casado 的博士论文题目是《一种名为 Ethane 的网络架构》，该论文得到了 Nick McKeown 、Scott Shenker 的帮助，2007 年，这三人成立了 Nicira。而这一年正好是是软件定义网络的元年。

SDN 追求的目标是更好地控制网络，即把网络控制模块化，对网络控制平面进行适当的软件抽象，使之可演进。

SDN 的第一个突破就是 OpenFlow，这是一种远程控制交换机和路由器的标准方式。可以理解为一种通用语言或指令集，有了它就可以编写网络的控制程序而无需为每一台路由器重写代码。

OpenFlow 的大旗一扯，众人即集结在其鞍前马后，其中不乏业界的大佬，如 Google、HP、NEC、爱立信等，众多媒体都把它称为是将网络带出野蛮的中世纪的救主。

不过有一点比较麻烦，如果硬件供应商不把 OpenFlow 加入自己的设备的协议支持清单，那么 OpenFlow 就无法在交换机和路由器上使用，就只能叫好不叫座。从结构和标准的角度来看，OpenFlow 无疑是重要的，但是从架构的角度而言却并非如此，因为更重要的组件关心的是协调交换机的活动，以便行为能够连贯协调。

OpenFlow 的终极目标不是更好地管理网络硬件，而是创建一种无需关心硬件即可组网的软件架构，也就是说实现组建虚拟网络的能力。

所以这家公司决定推出自己的解决方案—网络虚拟化平台。

网络虚拟平台（NVP）

Nicira 经常被用来跟 VMware 类比，大家都是从斯坦福的研究孕育而出的。大家做的都是虚拟化，后者是服务器的虚拟化，前者则是网络的虚拟化。虚拟层可以屏蔽底层硬件的复杂性。

VMWare、Xen 和 KVM 等也有虚拟化的努力，也实现了虚拟交换机，不过这些虚拟交换机功能有限，无法将多个虚拟交换机互连起来形成一个复杂的虚拟网络。

而 Nicira 的做法是在物理硬件之上加载一个虚拟网络平台，这个平台由虚拟交换机 Open vSwitch 和控制器组成。这个虚拟交换机是开源的，可以对任何人开放使用，虚拟交换机之间的连接由控制器进行管理。

Nicira 的解决方案是解耦、独立、控制。

（一）解耦

所谓解耦是指将网络的控制从网络硬件中脱离出来，交给虚拟的网络层处理。这个虚拟网络层加载在物理网络之上，在一个虚拟的空间重建整个网络。有了网络虚拟化，物理网络被泛化为网络能力池，正如服务器虚拟化把服务器设备转化为计算能力池。如图 2-1。

网络虚拟化使得 IP 连接大为简化，对物理网络的要求也大幅降低，二层网络的复杂性不复存在，VLAN 变得无关紧要。

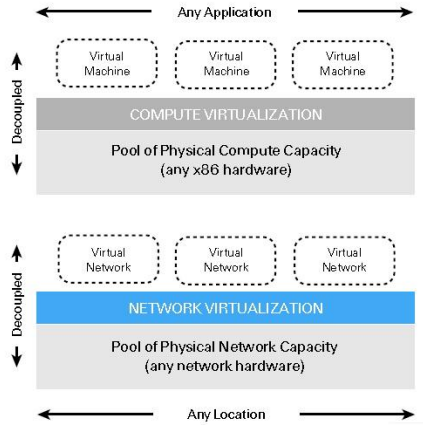


图 2-1

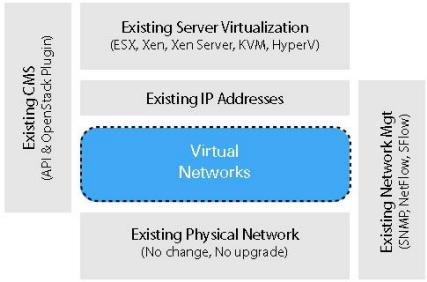


图 2-2

自由、奔放的技术刊物

（二）独立

加载的这层虚拟层无需现有的框架做出任何改变，原有网络硬件、原有的服务器虚拟化解决方案、原有的云管理系统、原有的 IP 都不需要改变。如图 2-2。

（三）控制

NVP 有两类关键组件：

一个是 Open vSwitch，这个东西是一种可以远程控制的交换机软件。其部署方式有两种，一种是部署在服务器的管理程序（Hypervisor）内，另一种是 NVP 网关，这个通常用来集成遗留的物理网络。

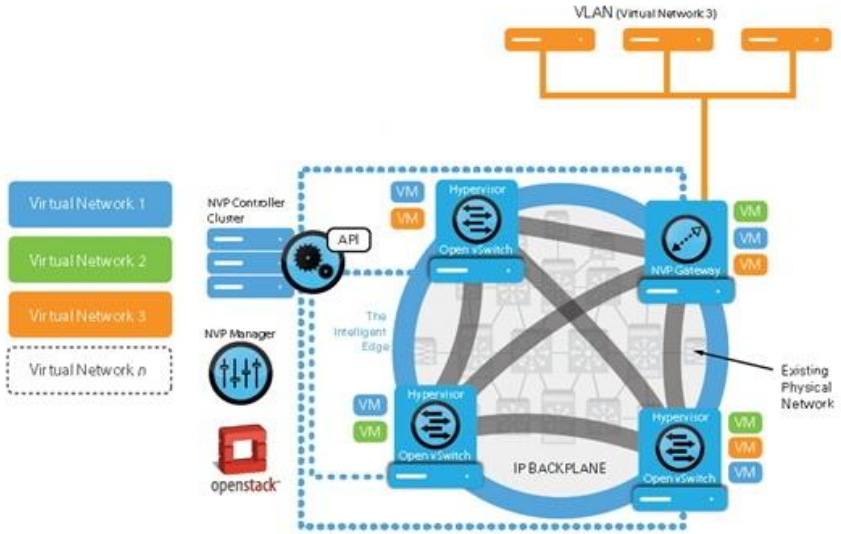
其二是控制器集群，运行在服务器上，管理所有的网络组件和连接。

从商业上看，其最大的优势是不需要底层硬件支持 OpenFlow，这样不管硬件厂商如何抵触也无法阻止网络虚拟化的脚步。

如果你不能打败敌人，那就加入他们。旧体系的霸主思科似乎也看清了这一点，目前也正在开发类似的技术，服务器虚拟化的巨头 VMWare 也要染指。不过 Nicira 坚信自己领跑的优势明显。

虚拟化一切

Nicira 的平台对于 Rackspace 这样的机构尤其合适。Rackspace 的定位是基础设施云（IaaS），为全球众多的开发者和企业提供虚拟服务器和存储服务，有了虚拟网络



的支持，Rackspace 可以把自己的客户限定在不同的虚拟网络中，保证安全。

但是 Yahoo 前 CTO Raymie Stata 质疑其对于单租户大网络的客户意义不大。比方说 Facebook，虽然其网络很大，但是自身并不需要划分那么多的虚拟网。

Casado 不同意这种说法。他说大公司的内部运营很复杂，尽管公司内部的网络资源不需要与外部共享，但是共享还是存在的，那就是内部众多的不同应用。应用不一样，部门不一样，对网络的需求也不一样，但是网络资源必然是共享的。像 Google 这样的公司，其运作模式跟 Racspace 和 Amazon 不会有太大差别，只不过前者是私有云，后者是公有云罢了。云者，其实就是大规模资源不受时间、地点、大小限制的灵活调配，只不过之前的云只有计算和存储的虚拟化能力，但是网络的灵活调配是个短板，现在 Nicira 出来补缺了。

网络设备商必将日渐式微

现在，许多最大型的互联网公司，包括 Google 在内，已经开始自造网络。他们的做法是向中国和台湾地区的制造商购买一些廉价的网络引擎，从而绕开思科和 Juniper 这样的耗钱大户，此举导致了网络商品化和芯片批发商的逐渐兴起。而现在，有了网络虚拟化平台的支持，由于底层硬件的无关性，这股趋势只会延续扩大下去。

思科也在开发自己的网络虚拟化工具，思科跟 Nicira 一样加入了 OpenStack 的网络虚拟框架的建设工作。（OpenStack 是建设基础设施云的一个开源平台。）

不过 Casado 不看好传统网络供应商愿意全心投入网络虚拟化。因为这就好像是搬起石头砸自己的脚。看看历史，有哪个占优势地位的公司愿意做一个可以兼容其他厂家任何类型产品的东西呢？他们也许会做做样子，但是具体的实际意义的事情肯定不会干。

Casado 相信传统的网络硬件上必将日渐式微。思科和 Juniper 等，会慢慢变得无足轻重。

最近 Casado 收到了一封电子邮件，来信者是 Nicira 过去有过往来的一家大公司的一位主管，此人询问 Casado 能否跟他碰个面。Casado 以为对方想进行公司合作，就答应了。结果此人原来是一位听说了 Nicira 的普通网络硬件商，约 Casado 出来就是想知道他会不会在 10 年之内离开现在的工作。

轻松分析 DHCP

By Charlie

每一天，服务器所产生的日志数量都是惊人的，但是，当发生紧急状况时，IT 管理员往往都能够从这些海量日志中找到有价值的信息。

通常，在 Unix 下对日志进行查找使用的是 grep 之类的低效率方式，Windows 下则是在事件查看器中查找，或者对系统日志转换成 Syslog 等格式进行搜索。而 Splunk 使用了现代搜索引擎技术对日志进行搜索，同时提供了一个非常强大的 AJAX 式的界面展现日志，如同单机版的 GoogleAnalytics 一样简单易行，使我们的数据分析达到了所看即所得的境界。

当然，今天我们要做的是使用 Splunk 分析企业中 Windows DHCP 服务器的海量日志。

第一步，要想战胜敌人就要了解敌人的确切位置，现在我们确定 DHCP 审核记录日志是否启用：

启用 DHCP 服务器记录的步骤

1. 打开 DHCPMicrosoft 管理控制台(MMC)管理单元。

2. 在控制台树中，单击要配置的 DHCP 服务器。

3. 在“操作”菜单上，单击“属性”。

4. 在“常规”选项卡上，选择“启用 DHCP 审核记录”，然后单击“确定”。

随后我们再确定敌人的数量：在 Windows Server 中，DHCP 服务器日志文件默认配置为允许存储的最大日志文件数量是 7，一周内每天一个文件。DHCP 审核日志默认位于%windir%\System32\Dhcp

第二步，想要战胜敌人就要先就先了解敌人！“知己知彼，百战不殆。”今天 Splunk 分析的对象是 Windows DHCP 日志，那我们首先要了解 DHCP 日志的内容都是什么？查阅 Microsoft 官方文档可以很容易找到 DHCP 日志文件格式的描述。如表 3-1。

第三步，我们将启用各种招式、阵型，与敌人拉开阵势。现在 DHCP 日志文件还在 Windows 服务器上，怎么将日志实时地被 Splunk 监控起来将是一个问题。解决方法是将 DHCP 日志文件夹进行共享，并创建读取权限用户，在 Splunk 服务器上使用 cifs 的类型来 mount。cifs 是 Common Internet File System，简单来说就是用来通过网络进行文件的读写、共享等功能。

在 Splunk Server 上面运行：mount.cifs "//DHCP_Server_IP/dhcp_log"/mnt/dhcp/ -o user=username%"password",iocharst=utf8 这样在本地就可以具备日志文件的读取权限了，还需要注意的是如果文件夹名中含有空格，应将路径放在“中”。

现在万事俱备，只欠东风，下面就开始进入实质性阶段吧！首先我们查看一下进行监控的日志内容。如图 3-1。

可以看到日志中的内容包含几个重要字段，包括 ID、日期、时间、描述、IP 地址、是主机名、MAC 地址。所以当我们遇到 IP 地址定位的时候，可以直接搜索 IP 并

找到 MAC 地址。如图 3-2。

如果需要更多信息，可以通过在微软查询了解每个字段所代表的事件类型。如果将 DHCP 服务器配置为代表 DHCP 客户端执行域名系统(DNS)动态更新，可以使用 DHCP 审核日志监视 DHCP 服务器发给 DNS 服务器的更新请求以及 DNS 记录更新成功和失败事件。以下是用于 DNS 动态更新事件的事件 ID。如表 3-2。

至此，我们通过 Splunk 获取到 DHCP 日志内容并进行分析目的已经达成。如何发挥 Splunk 的全部功能，从海量数据中获得更多更有价值的信息，也需要我们不断充实并继续努力。

其实 Splunk 取名源自于 spelunking（洞穴探险运动），对于一个出色的数据分析工具来说，让数据挖掘工作的目标更具挑战性，使工作的过程变得轻松并且充满乐趣也是我们的美好心愿。

字段	描述
ID	DHCP 服务器事件 ID 代码
日期	DHCP 服务器上记录此项目的日期
时间	DHCP 服务器上记录此项目的时间
描述	有关此 DHCP 服务器事件的描述
IP 地址	DHCP 客户端的 IP 地址
主机名	DHCP 客户端的主机名
MAC 地址	可供客户端的网络适配器硬件使用的媒体访问控制 (MAC) 地址

表 3-1

事件 ID	描述
30	DNS 动态更新请求
31	DNS 动态更新失败
32	DNS 动态更新成功

表 3-2

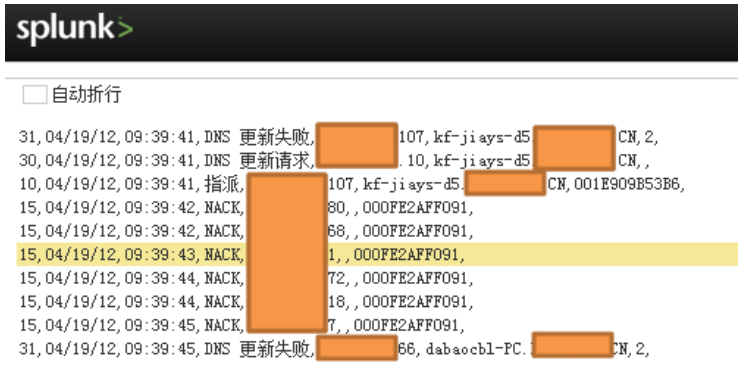


图 3-1

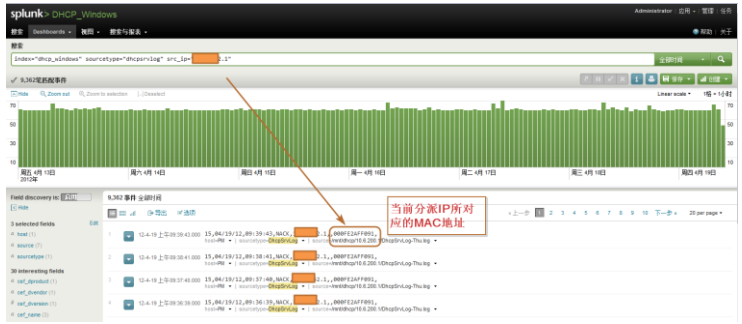


图 3-2

自由、奔放的技术刊物

Hi 大家好，我是 Simon。很高兴在《开源》上与大家见面。

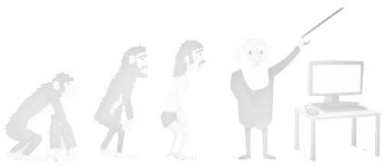
TCP/IP 是现代网络的基石，

不管您从事的工作是主机、数据库还是安全，亦或是一个智能手机使用者，您无时无刻不在和 TCP/IP 打交道。深入理解它能够对您的日常运维或应用开发带来很大的帮助。

从本期开始，我将每月和大家分享一些 TCP/IP 原理和实战中的经验。

如果大家有任何意见或建议，请随时和我联系。

Simon.fu@netis.com.cn



TCP 滑动窗口的简介

Simon（傅佳杰）：Netis 培训经理 5 年+ 网络和通讯领域工作经验。曾于华为任职多年。对 TCP/IP 原理有深刻认识和丰富的实战经验。

TCP 的滑动窗口主要有两个作用，一是提供 TCP 的可靠性，二是提供 TCP 的流控特性。同时滑动窗口机制还体现了 TCP 面向字节流的设计思路。

TCP 段中窗口的相关字段如图 4-1。

TCP 的 Window 是一个 16bit 位字段，它代表的是窗口的字节容量，也就是 TCP 的标准窗口最大为 2^16-1=65535 个字节。

另外在 TCP 的选项字段中还包含了一个 TCP 窗口扩大因子，option-kind 为 3，option-length 为 3 个字节，option-data 取值范围 0-14。窗口扩大因子用来扩大 TCP 窗口，可把原来 16bit 的窗口，扩大为 31bit。

滑动窗口基本原理

1）对于 TCP 会话的发送方，任何时候在其发送缓存内的数据都可以分为 4 类，“已经发送并得到对端 ACK 的”，“已经发送但还未收到对端 ACK 的”，“未发送但对端允许发送的”，“未发送且对端不允许发送”。“已经发送但还未收到对端 ACK 的”和“未发送但对端允许发送的”这两部分数据称之为发送窗口。如图 4-2。

当收到接收方新的 ACK 对于发送窗口中后续字节的确认是，窗口滑动，滑动原理如图 4-3。

当收到 ACK=36 时窗口滑动。

2）对于 TCP 的接收方，在某一时刻在它的接收缓存内存在 3 种。“已接收”，“未接收准备接收”，“未接收并未准备接收”（由于 ACK 直接由 TCP 协议栈回复，默认无应用延迟，不存在“已接收未回复 ACK”）。其中“未接收准备接收”称之为接收窗口。

发送窗口与接收窗口关系

TCP 是双工的协议，会话的双方都可以同时接收、发送数据。TCP 会话的双方都各自维护一个“发送窗口”和一个“接收窗口”。其中各自的“接收窗口”大小取决于应用、系统、硬件的限制（TCP 传输速率不能大于应用的数据处理速率）。各自的“发送窗口”则要求取决于对端通告的“接收窗口”，要求相同。如图 4-4。

滑动窗口实现面向流的可靠性

最基本的传输可靠性来源于“确认重传”机制。

TCP 的滑动窗口的可靠性也是建立在

自由、奔放的技术刊物

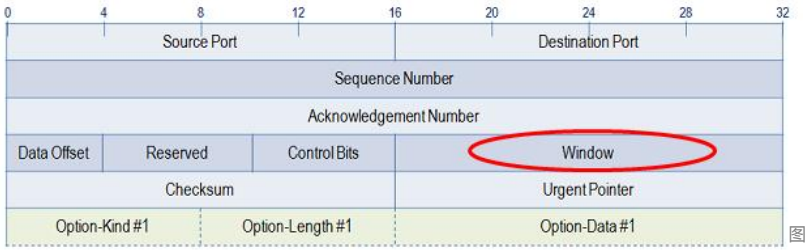


图 4-1

图 4-2

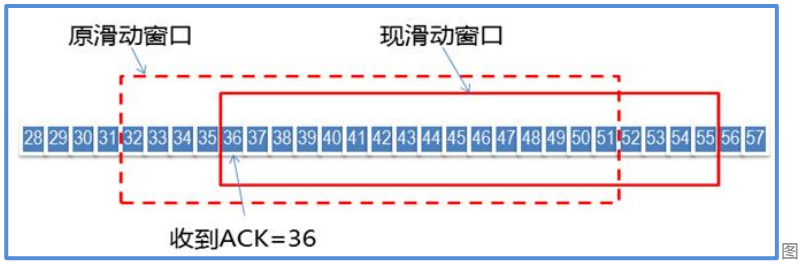


图 4-3

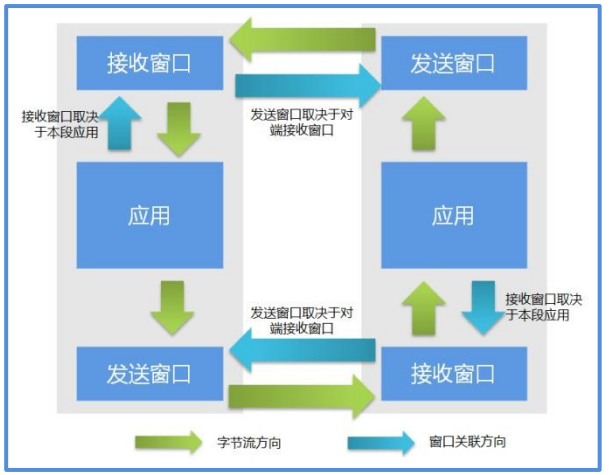


图 4-4

“确认重传”基础上的。

发送窗口只有收到对端对于本段发送窗口内字节的 ACK 确认，才会移动发送窗口的左边界。

接收窗口只有在前面所有的段都确认的情况下才会移动左边界。当在前面还有字节未接收但收到后面字节的情况下，窗口不会移动，并不对后续字节确认。以此确保对端会对这些数据重传。

滑动窗口的流控特性

TCP 的滑动窗口是动态的，我们可以想象成小学常见的一个数学题，一个水池，体积 V，每小时进水量 V1，出水量 V2。当水池满了就不允许再注入了，如果有个液压系统控制水池大小，那么就可以控制水的注入速率和量。这样的水池就类似 TCP 的窗口。应用根据自身的处理能力变化，通过本端 TCP 接收窗口大小控制来对对对端的发送窗口流量限制。

应用程序在需要（如内存不足）时，通过 API 通知 TCP 协议栈缩小 TCP 的接收窗口。然后 TCP 协议栈在下个段发送时包含新的窗口大小通知给对端，对端按通知的窗口来改变发送窗口，以此达到减缓发送速率的目的。

自由、奔放的技术刊物

在《TCP/IP 知识你问我答》环节，
将对大家在 TCP 学习中产生的问题做出答复。
如果您也有学习中的疑问，欢迎您与我联系
敬请关注。

TCP/IP 知识你问我答

问：TCP 的握手是 OSI 模型的那一层发起的？也就是 SYN SEND 这些包是哪一层发起的？

答：TCP 是属于 OSI 的第四层，传输层。SYN、FIN 等都是 TCP 段中的特征字符，都属于 OSI 的传输层。

最近一直有读者问到这样一个题目，我们来看一看：

问：在网络中设备发生拥塞时，TCP 可以使用滑动窗口技术很好的解决流量控制问题（）

A. Ture **B. False**

答：这个答案应该是 False。TCP 的滑动窗口的流控可以达到流控的目的，乍看下应该可以解决网络拥塞问题。但实际上并非如此，TCP 的发送窗口由对端的接收窗口控制，而对端的接收窗口只受限于对端的应用，与网络无关。不含 TCP 拥塞机制是，应用程序通过本段 TCP 协议栈根据自身需求通告对端窗口大小，在网络拥塞时一样可能通告较大窗口，导致 TCP 性能下降（重传过多）。

即使包含拥塞处理机制，TCP 对于拥塞的认知是靠发现重传超时。那么 TCP 被通告大窗口并在拥塞的网络中传输时，很容易丢包，于是启动慢启动，恢复后重复。这样会有间歇性的速率下降，会大大降低传输效率。并依然会间歇性的导致网络拥塞加重。